Diese Website verwendet Cookies von Google, um Dienste anzubieten und den Traffic zu analysieren. Deine IP-Adresse und dein User-Agent werden zusammen mit Messwerten zur Leistung und Sicherheit für Google freigegeben. So können Nutzungsstatistiken generiert, Missbrauchsfälle erkannt und behoben und die Qualität des Dienstes gewährleistet werden.

WEITERE INFORMATIONEN OK

Wednesday, November 11, 2009

Software and business method patents: at least four justices see through the Christmas ornament loophole

Several years ago, before section 101 of the U.S. patent statute became fashionable again, I wrote a paper on it, "Elemental Subject Matter." I remember several professors and patent attorneys, who shall remain nameless, telling me that section 101, which defines what kinds of subject matter are patentable and what kinds are not, was a useless topic to explore -- these issues, they said, had all been resolved and the legal excitement was elsewhere. I thought otherwise. I researched and in my paper I described the basic loophole that made software patents possible. Algorithms are "laws of nature" or "abstract ideas" and as such are supposed to be unpatentable. Patent lawyers being clever got around this by tacking on an extra fig-leaf or Christmas-ornament element to patent claims: the patent was for process X,Y,Z "and a computer", where X and Y and Z is the novel and non-obvious algorithm and "computer" is just your general-purpose computer. Under a long line of high court precedents, starting with the old English case of *Nielson v. Harford*, and continuing through many Supreme Court cases, this was an invalid claim: {X,Y,Z}, the part of the patent that makes it novel and non-obvious, must itself be patentable subject matter, i.e. not just an algorithm or law of nature or abstract idea. But the Federal Circuit, which hears all U.S. patent appeals and thus dominates U.S. patent law, ignored *Nielson*. Software became patentable because lawyers could trivially tack on "computer" or "memory" onto software claims, turning abstract algorithms into patentable "machines." Still later, the Federal Circuit allowed even these fig-leafs to be dropped from software patents, they were implicitly understood. The issue has never come before the U.S. Supreme Court. Until now.

At least four Supreme Court justices brought up the issue in Monday's oral arguments in *Bilski v. Kappos*, a business methods patent. The main patent claim reads as follows:

> A method for managing the consumption risk costs of a commodity sold by a commodity provider at a fixed price comprising the steps of:
>
> (a) initiating a series of transactions between said commodity provider and consumers of said commodity wherein said consumers purchase said commodity at a fixed rate based upon historical averages, said fixed rate corresponding to a risk position of said consumer;
>
> (b) identifying market participants for said commodity having a counter-risk position to said consumers; and
>
> (c) initiating a series of transactions between said commodity provider and said market participants at a second fixed

Pages

- Home

About Me

Nick Szabo

"A premier thinker about history, law and economics, and the lessons they have for security." -- Adam Shostack, Emergent Chaos

"Szabo comes out with these essays that leave me in awe." -- Brian Dunbar

"Reading material that is eclectic, challenging and endlessly fascinating." -- Sean McGrath, Propylon

"Like most blogs worth my attention, this blog is updated only infrequently. That is because the authors of blogs worth my attention only post when they have something to say that is true, relevant and not already known by their audience. Most of the human race does not have the skill to know when an idea has these three properties. The skill is particularly rare in the fields of politics and economics, which is why this blog is such a rare and valuable thing." -- Richard Hollerith

View my complete profile

rate such that said series of market participant transactions balances the risk position of said series of consumer transactions

(Forget about the fact that this is not even novel much less non-obvious. When the Federal Circuit allows claims to be made in areas where they previously weren't, the U.S. Patent Office agents are incompetent to analyze techniques in the new area or to search for prior art, and indeed a search of prior patents, which is almost all they know how to do, naturally turns up no prior art. Thus the many preposterously obvious software and business method patents we've seen. The case is being heard on the assumption that the patent office agent was correct, absurd as it is, to declare this claim novel and non-obvious, and the issue is thus focused on whether such business methods are patentable subject matter under section 101 of the patent code).

These four justices seem to agree with the view of my paper that the Christmas ornament loophole lies at the heart of software and business method patents:

JUSTICE STEVENS: I don't understand how that can be a patent on a machine if the only thing novel is the process that the machine is using. Isn't -- isn't the question -- really, the question there was whether the new process was patentable.
(p. 42)

(in reply to Justice Stevens repeating the above point)
JUSTICE KENNEDY: That's -- that's a problem I have.
(p. 44)

JUSTICE BREYER: But then all we do is every example that I just gave, that I thought were examples that certainly would not be patented, you simply patent them. All you do is just have a set of instructions for saying how to set a computer to do it. Anyone can do that. Now, it's a machine. So all the business patents are all right back in...all you do is you get somebody who knows computers, and you turn every business patent into a setting of switches on the machine because there are no businesses that don't use those machines.
(p. 46)

This is also what Chief Justice Roberts is clumsily getting at on pg. 35:

CHIEF JUSTICE ROBERTS: ...that involves the most tangential and insignificant use of a machine. And yet you say that might be enough to take something from patentability to not patentable.

I'd like to think that somebody over there in the Supreme Court building has been reading my paper, but more likely, yet remarkably, Justice John Paul Stevens, the author of *Parker vs. Flook*, the last case to apply *Nielson v. Harford* properly, and the only justice left from that 1977 court, still remembers *Nielson* and has taught a whole new generation of justices its meaning.

The implications of this view may seem harshly radical (if you rely on software patents) or pleasantly reactionary (if you fondly remember the days when we didn't have them). The patent bar and software patent holders have been in a tizzy since Monday, fearing that the Court's hostility to business method patents will lead to a ruling that will spill over to invalidate the recent non-ornamented software patents they have been drafting and the USPTO has negligently been approving. And software engineers have been dreaming that they will finally be freed from some of the increasingly dense patent thicket. But if the Court, as the above comments suggest, returns to *Nielson*, the result could be even more dramatic than is hoped or feared. Taking the *Nielson* logic to its conclusion would invalidate practically all software-only and business method patents, including ornamented ones. Those who want software patents would have to go do what they should have done in the first place -- get Congress to pass a statute expanding patentable subject matter to software, and very importantly command the USPTO to recruit and train computer scientists and people who know how to search the non-patent software literature for prior art so that software claims that don't make sense won't pass muster. Then, if this

Blog Archive

- ► 2018 (1)
- ► 2017 (3)
- ► 2016 (4)
- ► 2015 (3)
- ► 2014 (3)
- ► 2013 (3)
- ► 2012 (8)
- ► 2011 (12)
- ► 2010 (9)
- ▼ 2009 (29)
  - ▼ November (3)
    - Software and business method patents: at least fou...
    - The auction and the sword
    - Commencing a history of Roman political and legal ...
  - ► October (6)
  - ► September (3)
  - ► August (1)
  - ► July (1)
  - ► June (3)
  - ► May (6)
  - ► April (3)
  - ► January (3)
- ► 2008 (55)
- ► 2007 (47)
- ► 2006 (130)
- ► 2005 (44)

experiment works, a few decades later try the same method for business patents. And if the experiment doesn't work, scrap software patents. At this point, the Federal Circuit's illegitimate experiment with software and business method patents is failing miserably. Let's hope the Supreme Court takes this opportunity to restore its old patent jurisprudence that the Federal Circuit so shamelessly flouted.

Posted by Nick Szabo at 9:59 PM

4 comments:

Ray said...

What I don't understand is why anyone thinks *any* patents are valid or enforceable if this reasoning were to take place.

The essentially part of nearly *any* process in the modern world can be performed by general purpose computers and general-purpose actuators.

A robot that welds up a car in a particular way is absolutely no less general purpose and absolutely no more "physical" than an LCD screen that spits out photons or a printer that spits out paper, or a router that moves electrons around.

Scientifically speaking, *all* patents are nothing but process patents in the modern world.

The consequence is that Congress would almost *certainly* actually expand patentable content to software if it were ruled against. Anything else is nonsense.

No one can patent just the bare algorithm even today, because you're always allowed to execute it in your head if you want (however long that might take). The only thing that can be patented is running it on a machine that has actual physical mechanical consequences.

8:05 AM

nick said...

Ray, the difference is that if some of the novelty and non-obviousness of a new robot is in the actual circuit design or mechanism design, that is a "machine", not just in its programming, and it's not contested that these are patentable. Also physical processes are patentable, such as new chemical reactions and reactors.

11:49 AM

Alrenous said...

So, was the extension of copyright to software done properly, as you describe?

3:41 PM

nick said...

My paper covered patents, not copyrights. OTTOMH/IIRC, some lawyers tried to argue that machine instructions weren't

copyrightable, but then it was pointed out that old player piano rolls could be copyrighted. So they didn't have much of a legal argument against copyrighting computer programs.

5:04 PM

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Simple theme. Powered by Blogger.